УДК 681.3.07

ИНСТРУМЕНТАЛЬНЫЙ КОМПЛЕКС ДЛЯ СОЗДАНИЯ ЭКСПЕРТНЫХ СИСТЕМ, ИСПОЛЬЗУЮЩИХ МОДЕЛИ ФУНКЦИОНАЛЬНЫХ ОТНОШЕНИЙ

М.П. Силич, Н.Ю. Хабибуллина

Томский университет систем управления и радиоэлектроники
E-mail: silich@fet.tusur.ru

Описывается инструментальный комплекс «WinESISP», предназначенный для создания исследовательских прототипов экспертных систем в статических проблемных областях, использующих в качестве формы представления знаний модель функциональных отношений. Комплекс позволяет создавать модели, описывающие различные виды зависимости между параметрами предметной области (продукции, аналитические формулы, процедуры), осуществлять нечеткий логический вывод с помощью алгоритмов прямого и обратного вывода для решения задач диагностики, оптимального проектирования, планирования и управления системами.

Введение

Одной из современных тенденций в построении баз данных экспертных систем является комбинирование различных форм представления знаний. Модель функциональных отношений [1–3] позволяет наряду с правилами-продукциями использовать аналитические формулы и разнообразные процедуры-функции за счет структурирования исполняемых утверждений посредством задания сети отношений. Основными достоинствами данного способа организации операционного знания помимо возможности комплексирования разнородных фрагментов знаний являются: наглядность, «прозрачность» модели предметной области; простота внесения изменений; снижение трудоемкости поиска решений за счет упорядочивания вывода и «сужения» на каждом шаге множества проверяемых утверждений.

Для создания экспертных систем, использующих в качестве основы модель функциональных отношений был разработан ряд инструментальных средств – «МИС», «ЭСИСП», «ЭСФС», относящихся к классу оболочек [4–6]. С их помощью был создан ряд прототипов экспертных систем: «Тест-зрение» для экспресс-диагностики состояния зрения; «Тромбо-тест» для диагностики тромбоопасности по данным тромбоэлластограммы; «КЭНС» для проектирования оптимальных структур корреляционно-экстремальных навигационных систем; «Оргструктура» для выбора оптимальной организационной структуры системы управления и др. На базе прототипа «Тест-зрение» был создан коммерческий продукт путем его переработки с использованием процедурного языка программирования.

Опыт эксплуатации разработанных средств позволил выявить ряд новых требований, основными из которых являются: расширение возможностей нечеткого вывода (использование функций принадлежности, различных способов активизации и аккумуляции заключений правил); предоставление пользователю различных способов задания исходных данных (ввод в диалоге, загрузка из баз данных, генерирование случайных значений); возможность определения альтернативных способов логического вывода; графическое отображение функциональной сети зависимостей. Возникла потребность в разработке нового инструментария, удовлетворяющего выдвинутым требованиям.

Данная статья посвящена описанию инструментального комплекса «WinESISP» [7], ориентированного на язык функциональных отношений и предназначенного для создания исследовательских прототипов экспертных систем в статических проблемных областях для интерпретации данных, диагностики, оптимального проектирования и планирования. Комплекс позволяет описывать различные виды зависимости между параметрами модели (продукции, аналитические формулы, процедуры, в том числе процедуры фаззификации и дефаззификации, запросы и т.д.); осуществлять нечеткий логический вывод с помощью различных алгоритмов прямого и обратного вывода для решения прямой задачи (задачи интерпретации), обратной (задачи поиска допустимого решения) и задачи оптимизации. Кроме того, он обеспечивает объяснение полученных решений по протоколу вывода, формирование отчетов, визуализацию графической модели.

Модель функциональных отношений

Модель предметной области можно представить в виде:

$$M = \langle X, D, F \rangle,$$

где X – множество параметров (характеристик, свойств); D – множество доменов параметров (областей изменения значений); F – множество отношений функциональной зависимости между параметрами, каждое их которых связывает некоторый параметр-функцию с параметрами-аргументами и имеет смысл: «причина-следствие», «средство-цель», «ситуация-действие».

Графически модель функциональных отношений можно представить в виде направленной сети без циклов (рис. 1), вершинами которой являются параметры, а дугами – функциональные зависимости между ними.

В истоках сети находятся, так называемые, базовые параметры, значения которых не зависят от

значений других параметров. Остальные параметры непосредственно или опосредованно зависят от базовых.

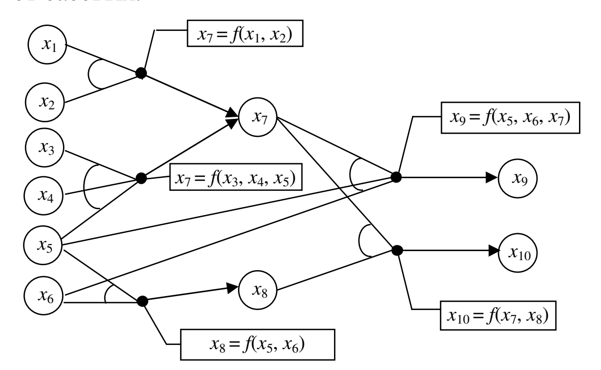


Рис. 1. *Сеть функциональных зависимостей параметров*

Структура классов базовых элементов модели представлена на рис. 2. Ключевыми элементами модели являются сущности «Knowledge base» (База знаний), «Parameter» (Параметр) и «Dependence» (Зависимость). База знаний задается с помощью атрибутов, содержащих идентификатор базы знаний (IDBase), ее имя (NameBase) и имя директории (PathFile). База знаний состоит из множества объектов класса «Parameter» и связанных с ними объектов других классов.

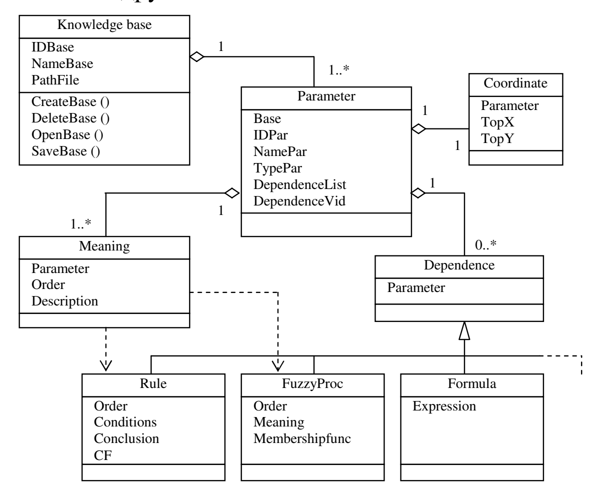


Рис. 2. *Диаграмма классов модели функциональных отношений*

Каждый параметр описывается набором атрибутов, среди которых наиболее важными являются: IDPar – идентификатор параметра, NamePar – его имя, TypePar – тип (количественный или лингвистический), DependenceList – список идентификаторов параметров, от которых данный параметр непосредственно зависит, DependenceVid – тип зависимости (правила-продукции, формула, процедура). Для описания отдельных значений из множества возможных значений параметра (домена) используется класс «Meaning» (значение), содержащий атрибуты: Parameter – идентификатор параметра, к которому относится значение; Order – порядковый номер значения; Description – описание (наименование значения для лингвистических параметров или диапазон принимаемых значений для количественных параметров). Для описания координат параметра в графической модели используется класс «Coordinate».

Класс «Dependence» для описания вида зависимости параметра-функции от параметров-аргументов имеет ряд потомков, соответствующих различным типам зависимостей. Каждый их этих классов включает атрибут Parameter, содержащий идентификатор параметра-функции. Состав остальных атрибутов различен. Класс «Formula» (Формула) включает в себя атрибут Expression, содержащий математическое выражение. Выражение может включать текущие значения параметров-аргументов, константы, знаки арифметических операций и математических функций и т.д.

Класс «Rule», соответствующий правилу-продукции, содержит: порядковый номер правила (Order), условия правила (Conditions), заключение (Conclusion) и коэффициент уверенности правила (CF). Условия представляют собой логические формулы для проверки значений параметров-аргументов. Каждое из условий соответствует одному из параметров. Заключение может содержать конкретное значение параметра-функции или формулу для вычисления значения. Для описания зависимостей типа процедур используются различные классы. Так, процедуре фаззификации сопоставляется класс «FuzzyProc», включающий: описание функции принадлежности (Membershipfunc); лингвистическое значение, которому соответствует данная функция (Meaning) и порядковый номер (Order).

Структура комплекса «WinESISP»

Инструментальный комплекс «WinESISP» имеет структуру, типичную для оболочек, и включает в себя следующие основные блоки: блок хранения, редактор базы знаний и блок вывода (рис. 3). Каждый из этих блоков в свою очередь содержит ряд подблоков. Большинство функциональных блоков реализовано в виде автономно работающих, взаимодействующих между собой программных компонентов.

Блок хранения включает базу знаний, базу фактов и протокол вывода. База знаний содержит постоянную информацию о параметрах и функциональных зависимостях, описывающих некоторую конкретную предметную область, в файлах формата СУБД Paradox 5.0. В базе фактов хранятся текущие значения параметров и факторы уверенности для этих значений, введенные пользователем и выведенные в процессе логического вывода. База фактов используется как рабочая область блоком вывода решений. В протокол вывода записывается последовательность номеров правил, активизированных в процессе вывода. Он используется блоком объяснений для формирования отчета о ходе поиска решения.

Редактор базы знаний служит для создания новых баз знаний, а также просмотра и редактирования существующих баз знаний. Содержит ряд специализированных редакторов – для ввода информации о па-

раметрах, для ввода правил-продукций, формул, построения функций принадлежности и т.д. Имеется графический редактор, позволяющий формировать сеть зависимостей параметров в виде графа.

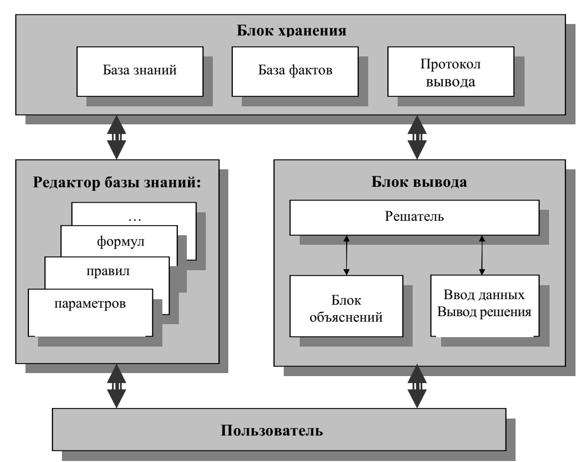


Рис. 3. *Структура комплекса «WinESISP»*

При редактировании осуществляется логический контроль вводимых данных. Так, при вводе формулы проверяется, совпадает ли количество открывающих и закрывающих скобок, не следует ли несколько знаков бинарных операций подряд и др. При вводе правил проверяется, входит ли вводимое значение параметра или интервал изменения значения в область допустимых значений. При формировании функции принадлежности проверяется, входят ли значения, по которым строится функция, в область допустимых значений соответствующего параметра и т.д.

Блок вывода решений реализует механизмы решения следующих задач: задачи интерпретации – поиска значения заданного параметра, соответствующего исходным значениям базовых параметров (истоков сети); задачи поиска допустимого решения – определения значений базовых параметров, соответствующих заданному значению целевого параметра; задачи оптимизации – поиска комбинации значений базовых параметров, обеспечивающей наилучшее значение критерия эффективности при заданных ограничениях.

Для решения задачи интерпретации используется алгоритм прямого вывода на сети функциональных отношений, для поиска допустимого решения – алгоритм обратного вывода на сети функциональных отношений, для нахождения оптимального решения могут использоваться оба алгоритма [2, 3]. Основное отличие алгоритмов прямого и обратного вывода на сети функциональных отношений от традиционных алгоритмов прямого и обратного вывода на системах продукций заключается в том, что в них предварительно находится так называемый «путь вывода» – цепочка параметров, являющихся целями вывода на каждом шаге процесса вывода. Благодаря наличию сети зависимостей формирование такой цепочки не представляет трудности. В процессе вывода последовательно перебираются параметры из «пути вывода» и для каждого из них выполняется метод прямого или обратного вывода для соответствующей функциональной зависимости. В процессе как прямого, так и обратного вывода могут возникать ветвления. Для их организации используется стратегия «сначала вглубь». В результате определяются все возможные решения с различными коэффициентами уверенности, соответствующие заданным исходным данным.

Решение задачи оптимизации может осуществляться методом как прямого, так и обратного вывода. В первом случае генерируются все возможные комбинации значений базовых параметров и для каждой из них определяются значения параметров, на которые наложены ограничения и значение параметра, выбранного в качестве критерия оптимизации. Во втором случае осуществляется поиск комбинации значений базовых параметров, обеспечивающей наилучшее (на первой итерации) или ближайшее к наилучшему (на следующих итерациях) значение критерия. Выбор метода зависит от различных характеристик модели предметной области. Пользователю рекомендуется наиболее эффективный способ поиска решения для конкретной базы знаний.

В процессе решения любой из задач блок вывода сначала запрашивает у пользователя исходные данные. При этом данные могут вводиться в диалоге, либо загружаться из текстового файла или базы данных, либо генерироваться датчиком случайных чисел. После окончания процесса вывода пользователю выдаются найденные решения и объяснения, каким образом были получены данные решения (рис. 4).

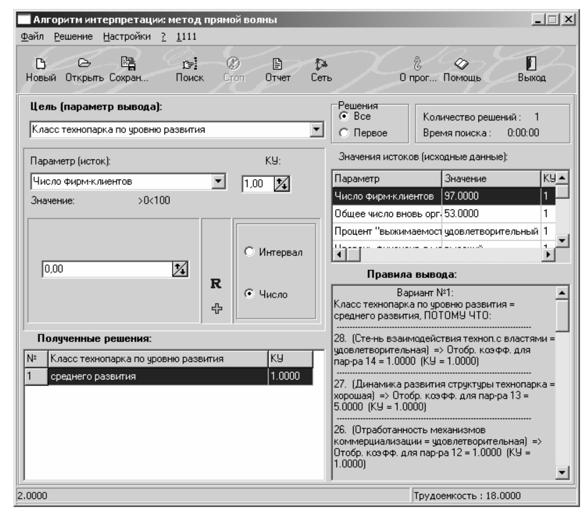


Рис. 4. *Экранная форма для решения задачи интерпретации*

Блок объяснений формирует объяснения по протоколу вывода. Пользователь может просмотреть всю цепочку параметров, задействованных в процессе решения – от истоков до заданного параметра. Для каждого параметра выдается текущее значение и коэффициент уверенности, а также правила, формулы или процедуры-функции, по которым они были выведены.

Найденные решения можно сохранить в текстовом файле или базе данных. Кроме того, может быть сформирован отчет о результатах вывода по заданной пользователем форме.

Заключение

Инструментальный комплекс «WinESISP» предоставляет непрограммирующему пользователю удобные средства для быстрого создания прототипов экспертных систем в статических предметных областях. Используемый данным инструментарием язык представления модели предметной области позволяет описывать различные виды зависимостей между параметрами модели. Встроенный механизм логического вывода может быть использован для решения задач интерпретации, поиска допустимого решения и поиска оптимального решения. Используемые при этом алгоритмы прямого и обратного вывода обеспечивают более высокую скорость вычислений по сравнению с традиционными алгоритмами прямого и обратного вывода на системе продукций за счет структурирования исполняемых утверждений.

Удобный пользовательский интерфейс и сервисные возможности (развитая справочная система, настройка опций, подсистема объяснений и формирования отчетов) облегчает пользователю создание базы знаний, ее тестирование и проведение экспериментов. Для создания коммерческой экспертной системы на базе прототипа, построенного средствами данного комплекса, можно использовать не только базу знаний, но и блок вывода решений, реализованный в виде отдельной COM-компоненты. Необходимо лишь разработать оригинальный интерфейс.

Комплекс «WinESISP» был использован для создания прототипа экспертных систем оценки финансово-экономического состояния предприятия на основе данных бухгалтерской отчетности, а также прототипа системы оценки уровня энергетической эффективности региона.

СПИСОК ЛИТЕРАТУРЫ

1. Силич М.П. Метод формирования гибридных моделей для построения экспертных систем // В сб.: Автоматическое и автоматизированное управление сложными системами. – Томск: Изд-во Том. ун-та, 1998. – С. 126–131.
2. Силич М.П. Системная технология: объектно-ориентированный подход. – Томск: Том. гос. ун-т систем управления и радиоэлектроники, 2002. – 224 с.
3. Силич М.П., Хабибулина Н.Ю. Поиск решений на модели функциональных отношений // Информационные технологии. – 2004. – № 9. – С. 27–33.
4. Силич В.А., Силич М.П. Использование "ЭСИСП" для создания экспертных систем и в обучении // В сб.: Автоматизация проектирования, идентификация и управление в сложных системах. – Томск: НТЛ, 1997. – С. 134–144.
5. Силич В.А., Хабибулина Н.Ю. Инструментальный программный комплекс создания гибридных экспертных систем, основанных на функциональных сетях // В сб.: Интеллектуальные автоматизированные системы проектирования, управления и обучения. – Томск: НТЛ, 2000. – С. 200–213.
6. Силич В.А., Силич М.П., Толмачев И.А., Хабибулина Н.Ю. Пакет программ для формирования гибридных моделей принятия решений на основе декларативной модели предметной области // В сб.: Автоматизация проектирования, идентификация и управление в сложных системах. – Томск: НТЛ, 1997. – С. 126–133.
7. Силич В.А., Силич М.П., Хабибулина Н.Ю., Чурсин А.В. Инструментальный комплекс «WinESISP». Свидетельство в Отраслевом фонде алгоритмов и программ № 3154 от 04.02.2004 г.

УДК 612.821.11.35